

1•VIEW

OneView: Setting up SSL/HTTPS

1	Introduction	3
1.1	Creating Keystore with Self-Signed Certificate.....	3
2	Creating OneView Keystore with Existing Certificate	4
3	Further documentation on Java and Tomcat SSL/HTTPS setup.....	4
4	Appendix 1: Export your Certificate and Private Key Using Windows.....	5
4.1	Start the Microsoft Management Console	5
4.2	Add/Remove Snap-in	5
4.3	Certificates Snap-in	5
4.4	Select Account.....	6
4.5	Export Certificate	8

1 Introduction

Using the SSL/HTTPS web protocol requires the use of a OneView keystore certificate. This document describes two different approaches to get started with the installation using HTTPS.

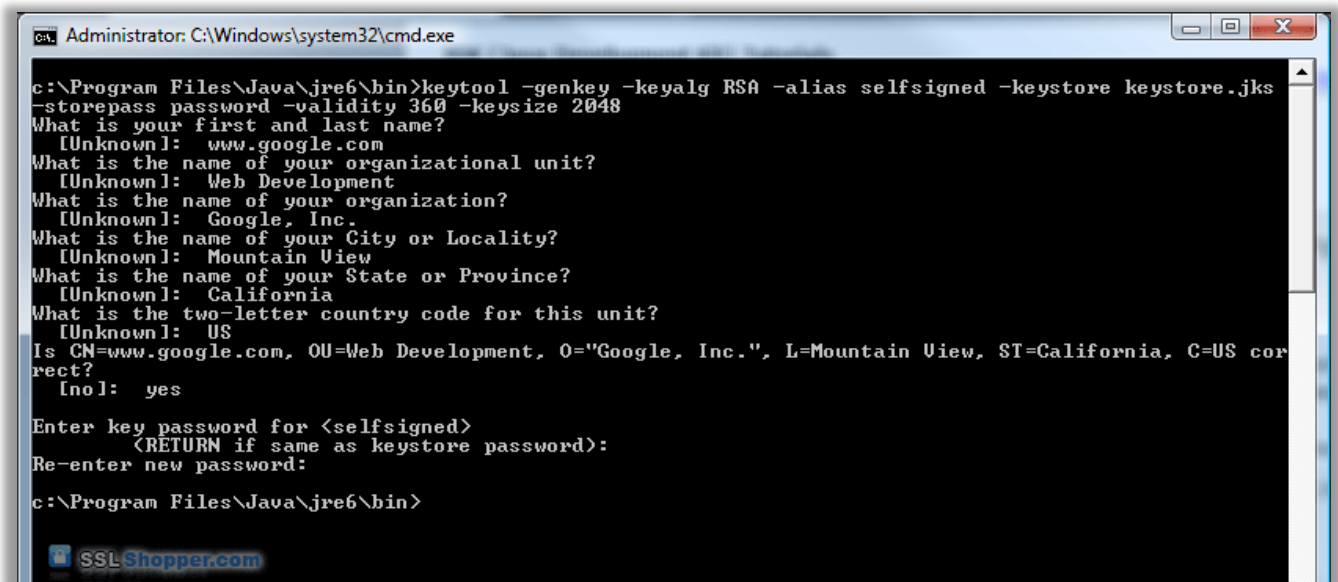
A few basic prerequisites are important to take note of prior to setting up HTTPS:

1. A keystore is always needed to activate HTTPS in OneView
2. A keystore can be created with a self-signed certificate or an existing certificate
3. A self-signed keystore can be used for the initial setup or if https is only used for extender setup
4. An existing certificate can be obtained from a Certificate Authority or exported from a Windows server certificate store

1.1 Creating Keystore with Self-Signed Certificate

The self-signed certificate can be generated by the use of a simple Java Keytool command:

1. Open the command console on whatever operating system you are using and navigate to the directory where standard Java keytool.exe is located (usually where the JRE is located, e.g. c:\Program Files\Java\jre6\bin on Windows machines).
2. Run the following command (where validity is the number of days before the certificate will expire):
`C:\Program Files\java\jre6\bin>keytool -genkey -keyalg RSA -alias selfsigned -keystore keystore.jks -storepass password -validity 360 -keysize 2048`
3. Fill in the prompts for your organization information. When it asks for your first and last name, enter the domain name of the server that users will be entering to connect to your application



```
Administrator: C:\Windows\system32\cmd.exe
c:\Program Files\Java\jre6\bin>keytool -genkey -keyalg RSA -alias selfsigned -keystore keystore.jks
-storepass password -validity 360 -keysize 2048
What is your first and last name?
[Unknown]: www.google.com
What is the name of your organizational unit?
[Unknown]: Web Development
What is the name of your organization?
[Unknown]: Google, Inc.
What is the name of your City or Locality?
[Unknown]: Mountain View
What is the name of your State or Province?
[Unknown]: California
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=www.google.com, OU=Web Development, O="Google, Inc.", L=Mountain View, ST=California, C=US cor
rect?
[no]: yes
Enter key password for <selfsigned>
<RETURN if same as keystore password>:
Re-enter new password:
c:\Program Files\Java\jre6\bin>
```

This will create a keystore.jks file containing a private key and a self-signed certificate. The next step is to configure the Java application to use the .jks file.

2 Creating OneView Keystore with Existing Certificate

1. Export key from server using e.g. the Windows guide in [Appendix 1](#).

Note: Do not extract the keys in other formats – all we need is the pfx-file (IISCERTIFICATE.pfx). Remember to specify a password for the exported key, which is required for step 2.

2. Create a new Java keystore with the following command:

```
"C:\Program Files\Java\jre1.8.0_60\bin\keytool.exe" -importkeystore -srckeystore C:\oneview\IISCERTIFICATE.pfx -srcstoretype pkcs12 -destkeystore C:\oneview\oneview\conf\OneView.keystore -storepass PASSWORD -deststoretype JKS
```

Match the keystore password in the oneview.conf file

3. Setup https settings in oneview.conf and restart the OneView Service

https.localhost.only	TLS Admin Web Server available locally only	false
https.keystore.filename	TLS Admin Web Server key store filename	conf/OneView.keystore
https.keystore.password	TLS Admin Web Server key store password	-

3 Further documentation on Java and Tomcat SSL/HTTPS setup

Follows standard Java and Tomcat keystore setup.

Please refer to the official documentation when installing and configuring SSL support on Tomcat and Oracle servers – helpful places to look could be:

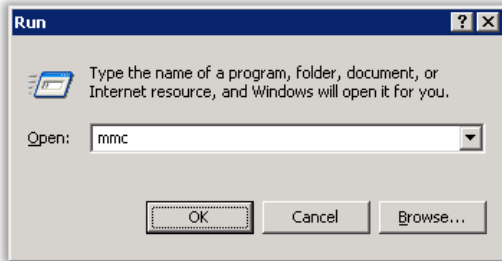
- <https://docs.oracle.com/en/java/>
- <https://tomcat.apache.org/>

Refer to documentation from your certificate provider for further information on your particular certificate and refer to Java/Tomcat for more information on certificate setup in general.

4 Appendix 1: Export your Certificate and Private Key Using Windows

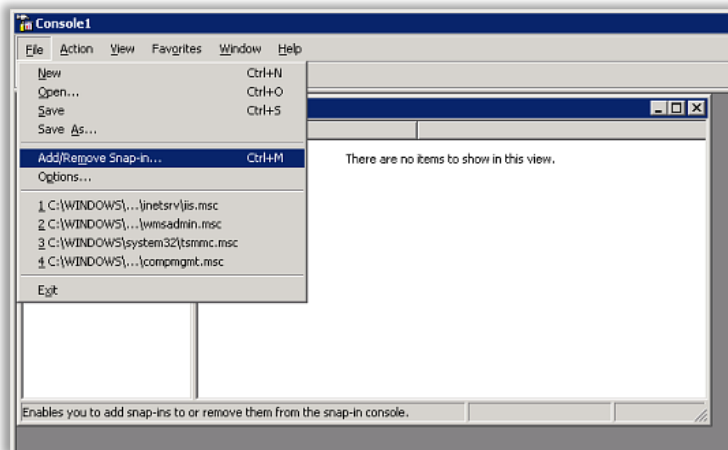
4.1 Start the Microsoft Management Console

Start the Microsoft Management Console by typing mmc in Run



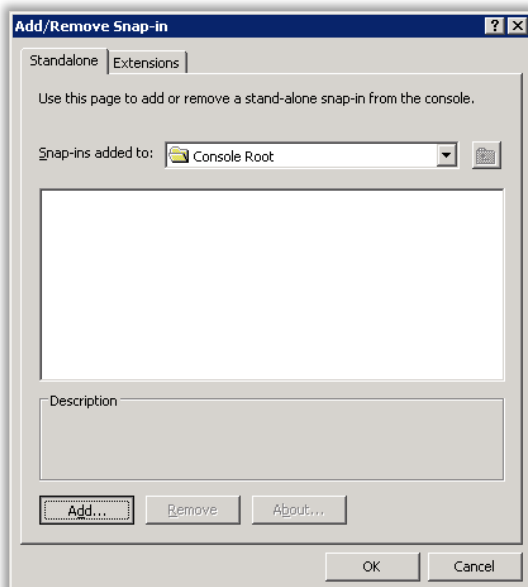
4.2 Add/Remove Snap-in

Click the 'Console' menu and then click 'Add/Remove Snap-in'.

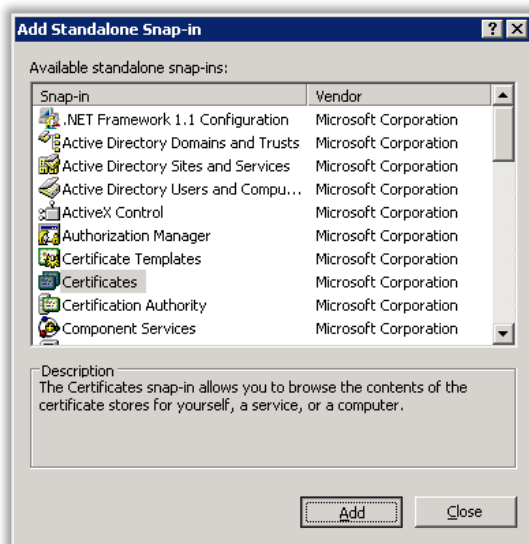


4.3 Certificates Snap-in

Click the 'Add' button and then choose the 'certificates' snap-in and click on 'Add'.

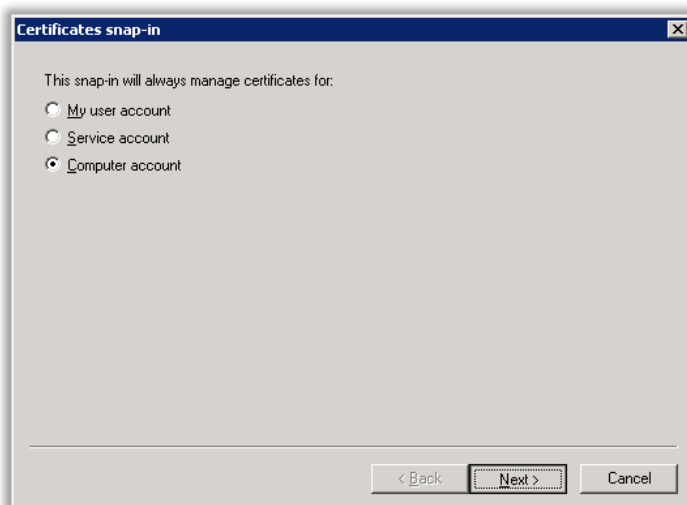


Depending on your Windows server installation you might be prompted with the next screenshot. Click 'Add' here to get to the next stage.

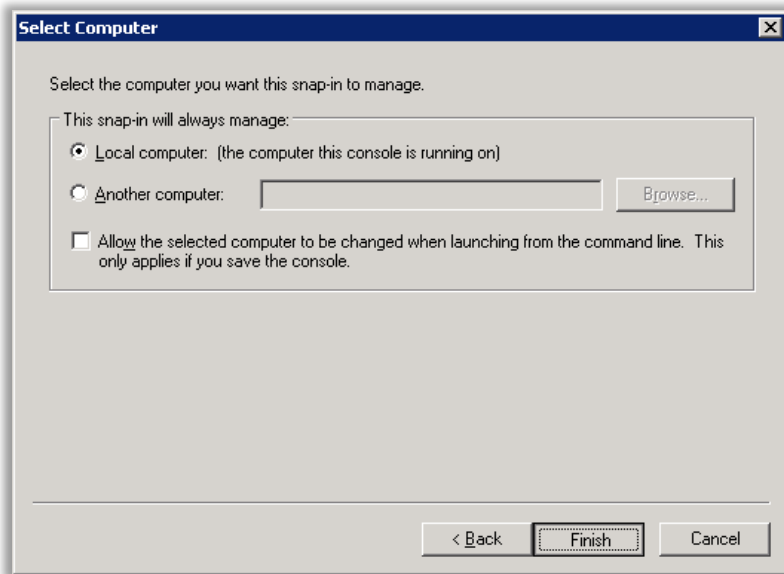


4.4 Select Account

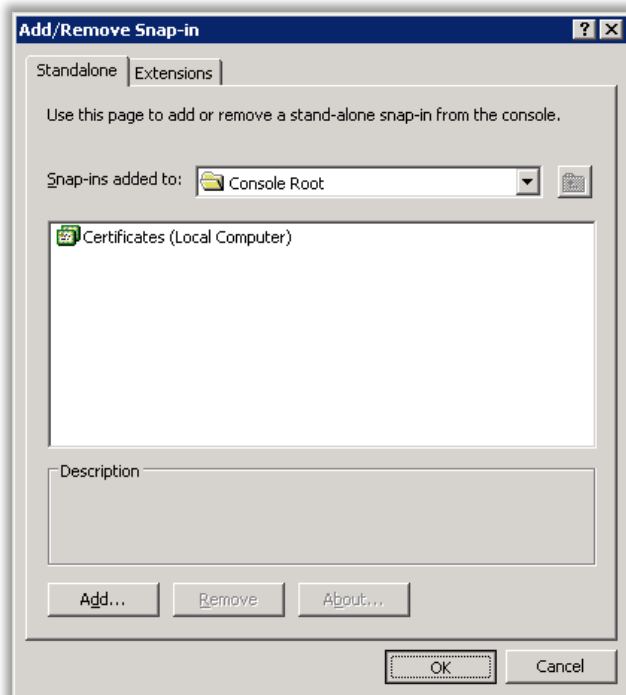
Select 'Computer Account' - then click 'Next'.



Select 'Local Computer' and then click 'Finish'

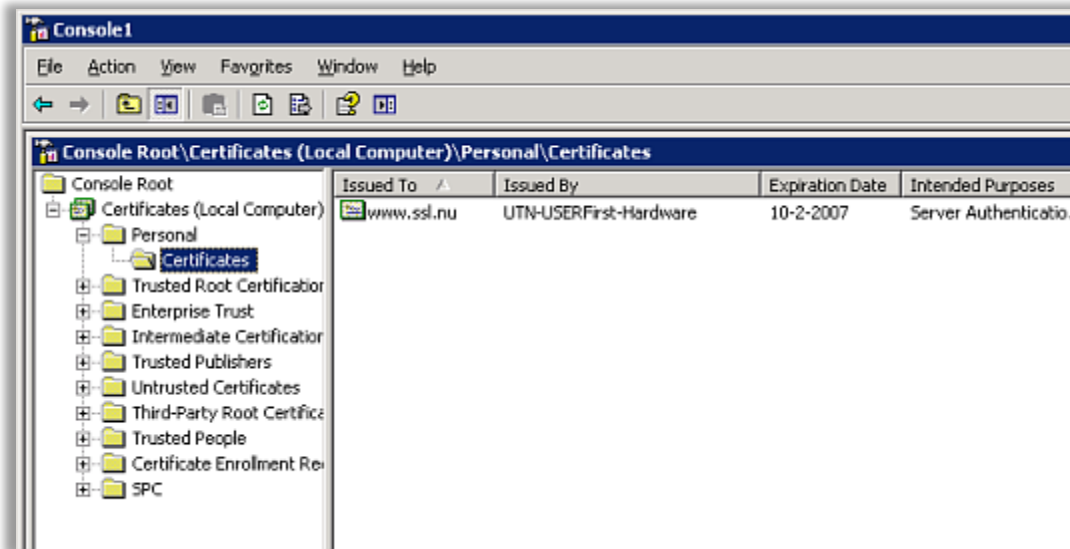


Click 'OK' to close the Add/Remove Snap-in dialogue

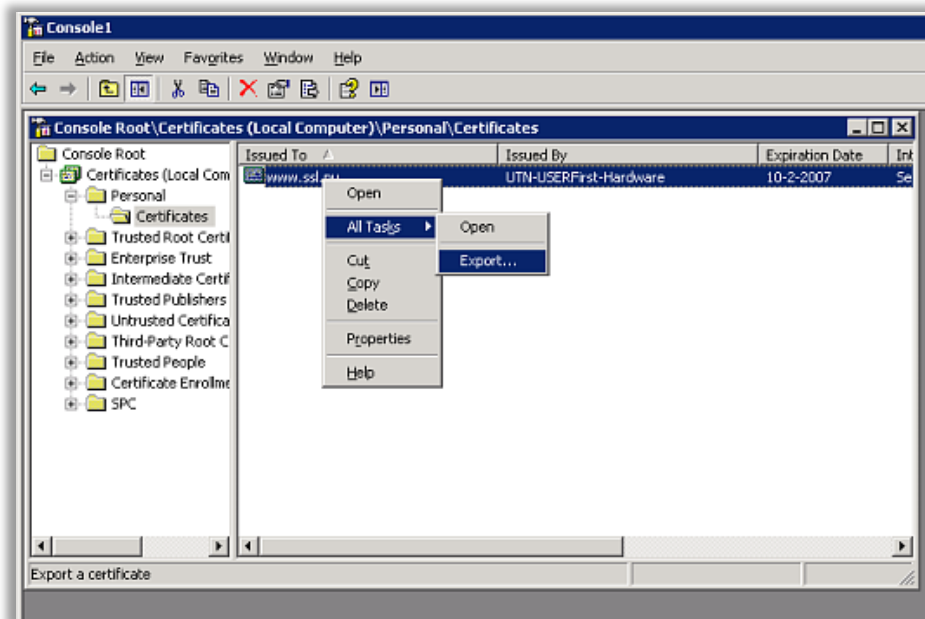


4.5 Export Certificate

Expand the menu for 'Certificates' – expand the 'Personal' folder and select 'Certificates'



Right click on the certificate that you want to export and select 'All tasks' > 'Export'



A wizard will appear. Make sure you check the box to include the private key and continue through with this wizard until you have a .PFX file.